REMARKS

Claims 1, 3-12 and 14-48 are pending in the present application. By this Response, claims 12, 14-20, 33, 34 and 46 are amended. Claims 12, 14-20, 33, 34 and 46 are amended to correct minor informalities. Reconsideration of the claims in view of the above amendments and the following remarks is respectfully requested.

I.   Application to be Considered Special

This application has received a fifth non-final Office Action. As per MPEP § 707.02, Applicants respectfully request that the Supervisory Patent Examiner personally check on the pendency of this application and make every effort to terminate prosecution.

II.   35 U.S.C. § 112, Second Paragraph

The Office Action rejects claims 20 and 46 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which applicants regard as the invention. Claims 20 and 46 are amended for clarity by providing proper antecedent basis and to more clearly recite the subject matter which Applicants regard as the invention. Therefore, the rejection of claims 20 and 46 under 35 U.S.C. § 112, second paragraph is overcome.

III.   Objection to Claims

The Office Action states that claims 14-19 are objected to because the first line of claims 14-19, ""The apparatus of claim" are not clearly. They should go with the system of claim 12." Thus, claims 14-19 have been amended to correct the minor informalities to overcome this objection.

IV. <u>35 U.S.C. § 103, Alleged Obviousness, Claims 1, 3-4, 7-12, 14-15 and 17-19</u>

The Office Action rejects claims 1, 3-4, 7-12, 14-15 and 17-19 under 35 U.S.C. § 103(a) as being unpatentable over Ng et al. (U.S. Patent No. 6,385,618 B1) in view of Text Book: Fundamentals of Database System (Third Edition) of Ramez Elmasri and Shamkant B. Navathe from Addison-Wisley and further in view of Sarkar (U.S. Patent 6,418,448 B1). Because this rejection is essentially the same as in the Previous Office Action, this rejection is respectfully traversed for the same reasons stated in the previous Response filed March 24, 2004, the remarks of which are hereby incorporated by reference. This rejection is respectfully traversed.

As to independent claim 1, the Office Action states:

> With respect to claim 1, Ng discloses determining a structure of the relational database (database schema of a relational database: col. 4, lines 23-27 and lines 35-36), wherein determining the structure of the relational database includes referring to a database meta-information class object associated with the relational database (database metadata where information of data concerning data, data definition, characteristics, relationships and external data a database of a database management system: see abstract, col. 7, lines 60-67 and col. 8, lines 1-18; also see fig. 9); and structure of the relational database as described in the meta-information class object (see fig. 5, the object-relational mapping tool is import database schema, which is containing the information of relationship objects and the class object of the database: col. 6, lines 3-67 and col. 7, lines 12-20).

> Ng discloses structure of relational database and schemas of relational database. Ng does not explicitly indicate determining a delete action based on the structure of the relational database and generating database modification commands based on the determined delete action and sending the database modification commands. Elmasri-Navathe discloses active database rules and triggers as referred to as the Event-Condition-Action or ECA-model for the delete operation such as a cascade deletion, the organization or structure of the tables have to be determine to in order to delete tuple that reference the tuple that is being deleted (see rule R4, TOTALSAL4 (page 737 and page 210). In combination, Ng and Elmasri-Navathe do not teach the relational database server in Java via JDBC interface.

> However, Sarkar discloses java classes are loaded in the database server (col. 11, lines 45-55; also see col. 6, lines 7-22).

> Therefore, it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Ng in view of Elmasri-Navathe with the teachings of Sarkar so as to

obtain database server of a object relational database locating of elements inside component relational schema with Java classes (col. 6, lines 13-15). This combination would provide a relational database having database server in the Java classes as argument for the interface of JDBC with SQL in the multi-tier client/server environment (Sarkar – col. 6, lines 20-28) and it is carrying an object SQL query for execution within one or more object relational schema (Sarkar – col. 6, lines 58-65 and querying and viewing multiple object relational schema in the large existing database (Sarkar – col. 7, lines 10-14) in the deletion of object in the relational database environment.

Office Action dated August 19, 2004, pages 3-5.

Claim 1, which is representative of the other rejected independent claim 12 with regard to similarly recited subject matter, reads as follows:

1.   A method of deleting object data from a relational database, comprising:
    determining a structure of the relational database, wherein determining the structure of the relational database includes referring to a database meta-information class object associated with the relational database;
    determining a delete action based on the structure of the relational database as described in the meta-information class object;
    generating database modification commands based on the determined delete action; and
    sending the database modification commands to a relational database server, wherein the relational database server deletes the object data from the relational database based on the database modification commands.

The Ng and Sarkar references have been discussed at length in the Responses filed July 16, 2003, December 2, 2003, and the March 24, 2004. In the March 24, 2004 Response, Applicant argued that none of the references teach or suggest referring to a database meta-information class object to determine the structure of a database and then determine a delete action based on the structure of the database determined from the meta-information class object, as recited in claims 1 and 12. The Office Action dated August 19, 2004 fails to provide rebuttal on Applicant's remarks with respect to claims 1, 3-4, 7-12, 14-15 and 17-19 from the Response dated March 24, 2004.

Ng teaches a system for updating an original object model, possibly having customizations from a programmer, with only the changes to the database made by a database administrator, as represented in the new database data structure. The result of

the update of the original object model is a combination of the original object model, any customizations added by the programmer, and the changes to the database made by the administrator. Once the original object model is updated, new source code is generated.

Sarkar teaches a method and apparatus for processing markup language specifications for data and metadata used inside multiple related internet documents. The method and apparatus of Sarkar are used to navigate queries and manipulate information from a plurality of object relational databases over the World Wide Web.

The newly cited Elmasri reference merely teaches that a delete operation may fail when the structure of a relational database is such that an entry that is being deleted is referenced by entries in other tables of the relational database. In such a case, the deletion may be rejected, a cascade delete operation may be attempted or the referencing attribute values that cause the failure may be modified (see page 210, section 7.3.2).

None of the references teach or suggest referring to a database meta-information class object to determine the structure of a database and then determine a delete action based on the structure of the database determined from the meta-information class object, as recited in claims 1 and 12. While Ng teaches using a DatabaseMetaData interface of JDBC to obtain database schema information and storing that database schema information in a data structure, such as data structure 700 in Figure 7 of Ng, there is no teaching or suggestion in Ng that a delete action is determined by obtaining structure information from the data structure and then determining a delete action based on the structure obtained from the data structure. To the contrary, Ng merely uses the database schema in the data structure to isolate changes between two database data structures, as shown in Figure 10 of Ng. This process includes determining if the number of tables have changed between database data structures, determining if the type, name or number of fields in the hash table of the two database data structures have changed, comparing primary keys for each table to determine if a different primary key has been designated as the primary key, and comparing foreign keys between both database data structures to determine if any of the foreign keys have changed. Based on these identified changes, an object model is updated for the relational database and source code is then generated based on the updated object model. (see column 7, line 13 to column 8, line 38).

Nowhere in Ng is there any teaching or suggestion that the database data structures that store the database schema are used to determine a structure of the database and then to determine a delete action based on the structure of the database obtained from the database data structure. To the contrary, if an object is to be deleted from a table in the relational database of Ng, the delete operation will be performed in a conventional manner, such as that taught by Elmasri. That is, the delete action will be attempted and if it fails due to an integrity violation, the delete action may be rejected, a cascade delete operation may be attempted, or referencing attribute values that cause the integrity violation may be modified (see Elmasri, page 210, section 7.3.2). This delete action is not determined based on database structure information obtained from a meta-information class object. To the contrary, the delete actions in both Ng and Elmasri are based on attempting delete operations on the relational database itself and determining if the operation fails. Ng and Elmasri, taken alone or in combination, fail to teach or suggest to use a database meta-information class object to determine a structure of a relational database and then determine a delete action based on the structure of the relational database determined using the meta-information class object.

Sarkar does not make up for these deficiencies either. While Sarkar may teach Java classes being loaded in a database server, Sarkar does not teach or suggest to identify a structure of a relational database by referring to a database meta-information class object associated with the relational database or determining a delete action based on the structure of the relational database determined from the database meta-information class object.

The Office Action alleges that the database meta-information class object is taught by Ng at column 7, lines 60-67, column 8, lines 1-18 and in Figure 9. These portions of the Ng reference are directed to the DatabaseMetaData interface of the Java Database Connectivity (JDBC) tool. While the DatabaseMetaData interface may be used to obtain database schema information, the DatabaseMetaData interface is a set of methods associated with the JDBC tools and is not a database meta-information class object itself that is associated with a relational database. To the contrary, the DatabaseMetaData interface methods form a tool that may be used to generate a database data structure that stores the database schema. Furthermore, the DatabaseMetaData

BEST AVAILABLE COPY

interface is not used in Ng to determine a structure of a relational database such that a delete action is determined based on the structure of the relational database determined via the DatabaseMetaData interface.

Thus, in view of the above, Applicant respectfully submits that Ng, Sarkar and Elmasri, taken alone or in combination, fail to teach or suggest determining a structure of a relational database by referring to a database meta-information class object associated with the relational database and determining a delete action based on the determined structure of the relational database as described in the meta-information class object, as recited in claims 1 and 12. At least by virtue of their dependency on claims 1 and 12, the specific features of dependent claims 3-4, 7-11, 14, 15 and 17-19 are not taught or suggested by Ng, Sarkar and Elmasri, either alone or in combination. Accordingly, Applicant respectfully requests withdrawal of the rejection of claims 1, 3, 4, 7-12, 14, 15 and 17-19 under 35 U.S.C. § 103(a).

In addition to the above, Ng, Sarkar and Elmasri, taken alone or in combination, fail to teach or suggest the features of dependent claims 3, 4, 7-11, 14, 15 and 17-19. For example, with regard to claims 4 and 15, Ng, Sarkar and Elmasri, taken alone or in combination, fail to teach or suggest that the database meta-information class object includes a delete action identifier for each dependent table of a plurality of tables in a relational database. The Office Action alleges that this feature is taught by Ng at column 3, lines 62-67, column 7, lines 60-67, column 8, lines 1-17 and in Figure 9 (see rejection of claims 4 and 7 on page 5 of the Office Action). Column 3, lines 62-27, reads as follows:
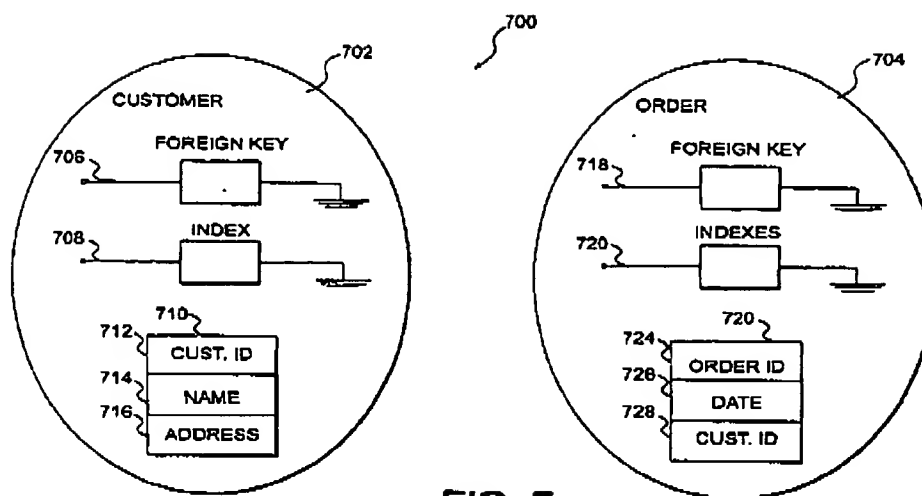
> The secondary storage device contains a database having a logical structure comprising tables with rows and columns. The memory contains a first database data structure reflecting the logical structure of the database and the object model containing objects based on the first database data structure.

This portion of Ng merely states that the database has tables with rows and columns and that the memory contains a database data structure that reflects the logical structure of the database and the object model. There is nothing in this section of Ng that teaches or even suggests a meta-information class object that includes <u>a delete action identifier for each dependent table of a plurality of tables</u> in a relational database.

Column 7, line 60 to column 8, line 17 of Ng, which describes Figure 9 of Ng, reads as follows:

FIG. 9 depicts a flowchart of the states performed when importing the database schema. Below, the object-relational mapping tool utilizes a number of methods which are found on the DatabaseMetaData interface of JDBC. The first state performed by the object-relational mapping tool is to call the GetTable method of the JDBC interface, which returns a description of the tables of the database (state 902). After retrieving this table information, the object-relational mapping tool selects one of the tables (state 904) and invokes the GetColumns method on the JDBC interface, returning a description of all of the columns in that table (state 906). Next, the object-relational mapping tool invokes the GetPrimaryKeys method to receive the primary key for the table (state 908). After obtaining the primary key, the object-relational mapping tool invokes the GetImportedKeys method to obtain information regarding the foreign keys (state 910). After invoking this method, the object-relational mapping tool determines if there are additional tables to be processed (state 912). If so, processing continues to state 904. Otherwise, the object-relational mapping tool constructs a database data structure, like the one shown in FIG. 7, from all of the information received in the previous states (state 914).

Nowhere in this portion of Ng is there any teaching or suggestion regarding including a delete action identifier, for each dependent table of a plurality of tables in a relational database, in a meta-information class object associated with the relational database. To the contrary, all this section of Ng teaches is the use of the various methods made available in the DatabaseMetaData interface of the JDBC to determine the structure of the relational database and then to use this information to generate the data structure shown in Figure 7, which is reproduced below:

FIG. 7

Conspicuously missing from this data structure 700 is any delete identifier. To the contrary, as shown above, the only elements of this data structure are objects 702 and 704, relation objects 706, 708, 718, and hash tables 710 and 720. Nowhere in the data structure depicted in Figure 7 is there any delete action identifier, let alone a delete action identifier for each dependent table of a plurality of tables in a relational database.

Thus, despite the allegations made by the Office Action, Ng does not actually teach the feature of a database meta-information class object including a delete action identifier for each dependent table of a plurality of tables in a relational database. Furthermore, as stated above, neither of the other references, Sarkar andmasri, teaches or suggests this feature either. Sarkar has nothing to do with delete action identifiers in meta-information class objects and is merely used to allegedly teach Java classes being loaded into a database server. Elmasri, while teaching that a cascade delete operation may be attempting when a delete action fails due to the entry being deleted also being referenced by other entries in other tables of the relational database, provides no teaching or suggestion with regard to including a delete action identifier, for each dependent table of a plurality of tables in a relational database, in a meta-information class object that is associated with the relational database. Since none of these references alone teach or

suggest this feature, any alleged combination of these references still would not result in this feature being taught or suggested.

Dependent claims 7-10 and 17-19 recite a file describing the structure and delete actions for tables in a relational database. These claims further define the file as being an Extensible Markup Language file, being generated based on user input to override default delete action identifiers in the file, and being generated based on user input to insert one or more delete constraints in the file for one or more of the tables in the relational database. None of these features are taught by Ng, Sarkar or Elmasri, either alone or in combination, because none of these references teach or suggest a file describing the structure and delete actions for tables in a relational database.

The Office Action alleges that these features are taught at column 3, lines 62-67, column 7, lines 60-67, column 8, lines 1-17, Figure 9, and column 7, lines 16-26 of Ng. Column 3, lines 62-67 of Ng reads as follows:

> The secondary storage device obtains a database having a logical structure comprising tables with rows and columns. The memory contains a first database data structure reflecting the logical structure of the database and an object model containing objects based on the first database data structure.

Nothing in this section of Ng teaches a file that describes the structure <u>and delete actions</u> for tables in a relational database. At most, the database data structure referenced in this section teaches a structure of the relational database. Nothing in Ng teaches a file that describes the structure <u>and delete actions</u> for tables in a relational database. The other sections of Ng reference by the Office Action merely describe the DatabaseMetaData interface of JDBC and the methods executed in the flow shown in Figure 9, which have been addressed in detail above. Moreover, since none of these sections of Ng teach or suggest a file that describes the structure and delete actions for tables in a relational database, Ng cannot teach that the file is an Extensible Markup Language file, the file is generated based on user input to override default delete action identifiers in the file, or that the file is generated based on user input to insert one or more delete constraints in the file for one or more of the tables in the relational database. None of the references teaches or suggests these features.

With regard to claim 16, this claim recites that the delete action identifier is one of cascade delete and nullify columns delete. While the prior art teaches these two different types of delete operations, nowhere in the cited art is there any teaching of a delete action identifier in a database meta-information class object that is one of a cascade delete and a nullify columns delete identifier, as recited in claim 16. Simply teaching these delete operations does not make obvious a delete identifier in a meta-information class object that is one of a cascade delete and a nullify columns delete identifier.

Thus, dependent claims 3, 4, 7-11, 14, 15 and 17-19 also distinguish over Ng, Sarkar and Elmasri, either alone or in combination, by virtue of the specific features recited in these claims.

V.   35 U.S.C. § 103, Alleged Obviousness, Claims 5, 6, 16 and 45

The Office Action rejects claims 5, 6, 16 and 45 under 35 U.S.C. § 103(a) as being unpatentable over Ng et al. (U.S. Patent No. 6,385,618 B1) in view of Text Book: Fundamentals of Database System (Third Edition) of Ramez Elmasri and Shamkant B. Navathe from Addison-Wisley and further in view of Sarkar (U.S. Patent 6,418,448 B1) and Crus et al. (U.S. Patent No. 4,947,320). Because this rejection is essentially the same as in the Previous Office Action, this rejection is respectfully traversed for the same reasons stated in the previous Response filed March 24, 2004, the remarks of which are hereby incorporated by reference. This rejection is respectfully traversed.

The Office Action dated August 19, 2004 fails to provide rebuttal on Applicant's remarks with respect to claims 5, 6, 16 and 45 from the Response dated March 24, 2004. In that Response, Applicant argued that the rejection is respectfully traversed for at least the same reasons as noted above with regard to claims 1, 12 and 43 from which claims 5, 6, 16 and 45 depend. Specifically, Ng, Elmasri and Sarkar, taken alone or in combination, fail to teach or suggest determining a structure of a relational database by referring to a database meta-information class object and determining a delete action based on the structure described in the meta-information class object, or a class object that is generated based on a structure of a relational database and one or more delete actions for tables in the relational database.

Moreover, Crus does not provide for the deficiencies of Ng, Elmasri and Sarkar. Crus teaches a delete set null and a delete cascade operation, as discussed in previously filed Responses. However, Crus provides no teaching or suggestion regarding a class object that is generated based on a structure of a relational database and one or more delete actions for tables in the relational database. Crus also provides no teaching or suggestion regarding determining a structure of a relational database by involving a meta-information class object associated with the relational database and then determining a delete action based on the determined structure of the relational database. Thus, even if Crus were combinable with Ng, Elmasri and Sarkar, the result still would not be the invention recited in independent claims 1, 12 and 43, from which claims 5, 6, 16 and 45 depend.

Furthermore, there is no teaching or suggestion in Crus to include a delete set null or a delete cascade operation identifier, <u>for each dependent table of a plurality of tables in a relational database</u>, in a meta-information class object, as recited in claims 5 and 16 or information identifying a delete set null or delete cascade operation in a class object, as recited in claim 45. While Crus may generally teach delete set null and delete cascade operations, there is nothing in Crus that teaches or suggests including information regarding such operations in a class object generated based on a structure of a relational database and one or more delete actions.

In view of the above, Applicant respectfully submits that none of the cited references, whether taken alone or in combination, teaches or suggests the features of independent claims 1, 12 and 43. At least by virtue of their dependency on claims 1, 12 and 43, the specific features of dependent claims 5, 6, 16 and 45 are not taught or suggested by the cited references, either alone or in combination. Accordingly, Applicant respectfully requests withdrawal of the rejection of claims 5, 6, 16 and 45.

VI. **35 U.S.C. § 103, Alleged Obviousness, Claims 20, 21, 24-28, 31-36, 39-43 and 46-48**

The Office Action rejects claims 20, 21, 24-28, 31-36, 39-43 and 46-48 under 35 U.S.C. § 103(a) as being unpatentable over Ng et al. (U.S. Patent No. 6,385,618 B1) in view of Text Book: Fundamentals of Database System (Third Edition) of Ramez Elmasri and Shamkant B. Navathe from Addison-Wisley. This rejection is respectfully traversed for the same reasons stated in the previous Response filed March 24, 2004, the remarks of which are hereby incorporated by reference. This rejection is respectfully traversed.

The Office Action dated August 19, 2004 fails to provide rebuttal on Applicant's remarks with respect to claims 20, 21, 24-28, 31-36, 39-43 and 46-48 from the Response dated March 24, 2004. In that Response, Applicant argued that with regard claims 20, 27 and 35, these claims recite generating a class object based on a determined structure and determined one or more delete actions. Ng and Elmasri, taken alone or in combination, fail to teach or suggest this feature. While Ng uses the DatabaseMetaData interface of JDBC to obtain database schema information which is then stored in a database data structure, Ng does not generate this database data structure based on one or more delete actions determined based on the structure of the relational database. To the contrary, as shown in Figure 9 and described in columns 7 and 8, the DatabaseMetaData interface merely gets a description of tables in the database, gets a description of the columns in the tables, gets the primary keys for the tables, and gets the foreign keys for the tables. Nowhere in Ng is there any teaching or suggestion to use the DatabaseMetaData interface to determine one or more delete actions based on the structure of a relational database and then generate a class object based on the determined one or more delete actions.

Elmasri does not teach or suggest these features either. Elmasri merely teaches attempting a cascade delete operation when a delete operation fails due to an integrity violation. Thus, Ng and Elmasri, taken alone or in combination, fail to teach or suggest the features of claims 20, 27 and 35. Therefore, since their dependent claims incorporate the subject matter of these respective independent claims, the dependent claims are also not taught or suggested by the alleged combination of Ng and Elmasri, either alone or in combination.

Regarding the remaining independent claims 43 and 46, these claims recite similar features to that emphasized above with regard to claims 20, 27 and 35. In particular, claim 43 recites a database meta-information generator class for generating a class object based on the determined structure and the determined one or more delete actions. Claim 46 recites generating a class object based on a determined structure, determined one or more delete actions, and user input. Thus, for similar reasons as noted above with regard to claims 20, 27 and 35, claims 43 and 46 define over the proposed combination of Ng and Elmasri.

In view of the above, Applicant respectfully submits that Ng and Elmasri, taken alone or in combination, fail to teach or suggest the features of independent claims 20, 27, 35, 43 and 46. At least by virtue of their dependency, the specific features of dependent claims 21, 24-26, 28, 31-34, 36, 39-42, 47 and 48 are not taught or suggested by the alleged combination of Ng and Elmasri. Accordingly, Applicant respectfully requests withdrawal of the rejection of claims 20, 21, 24-28, 31-36, 39-43 and 46-48 under 35 U.S.C. § 103(a).

In addition to the above, Ng and Elmasri, taken alone or in combination, fail to teach or suggest the features of dependent claims 21, 24-26, 28, 31-34, 36, 39-42, 47 and 48. For example, Ng and Elmasri, either alone or in combination, do not teaches or suggests the specific features recited in dependent claim 24. None of the cited references teach or suggest one or more delete actions being determined from a file describing the structure and delete actions for tables in the relational database, as recited in claim 24. The Office Action alleges that this feature is taught by Ng at the same portions discussed above and in previous responses. Again, there is nothing in Ng that teaches or suggests a file that describes the structure and delete actions for tables in a relational database. Ng teaches a data structure that identifies the structure of the relational database, however there is nothing in this data structure that identifies delete actions for tables in the relational database.

Additionally, Ng and Elmasri, taken alone or in combination, fail to teach or suggest a user input that overrides one or more default delete actions (claim 47) or inserts one or more delete action constraints (claim 48). The Office Action alleges that these features are taught by Ng at column 4, lines 45-67, column 6, lines 42-64 and column 7,

lines 9-67. Column 4, lines 45-67 merely teaches the incorporation of changes into an existing object model and the generation or source code. Column 6, lines 42-64 merely teaches methods in class 420 for getting and setting the values of data members and the use of a foreign key to create a relationship in source code. Column 7, lines 9-67 merely describes a process for a database administrator to add a column to a customer table. Nowhere in any of these sections, or any other section of Ng or Elmasri, is there any teaching or suggestion with regard to user input that overrides one or more default delete actions or inserts one or more delete action constraints.

Dependent claims 24-26, 31-34, and 39-42 recite a file describing the structure and delete actions for tables in a relational database. These claims further define the file as being an Extensible Markup Language file, being generated based on user input to override default delete action identifiers in the file, and being generated based on user input to insert one or more delete constraints in the file for one or more of the tables in the relational database. These features are not taught because none of the applied references teaches or suggests a file describing the structure and delete actions for tables in a relational database.

The Office Action alleges that these features are taught at column 3, lines 62-67, column 7, lines 60-67, column 8, lines 1-17, Figure 9, and column 7, lines 16-26 of Ng. Column 3, lines 62-67 of Ng reads as follows:

> The secondary storage device obtains a database having a logical structure comprising tables with rows and columns. The memory contains a first database data structure reflecting the logical structure of the database and an object model containing objects based on the first database data structure.

Nothing in this section of Ng teaches a file that describes the structure and delete actions for tables in a relational database. At most, the database data structure referenced in this section teaches a structure of the relational database. Nothing in Ng teaches a file that describes the structure and delete actions for tables in a relational database. The other sections of Ng reference cited by the Office Action merely describe the DatabaseMetaData interface of JDBC and the methods executed in the flow shown in Figure 9, which have been addressed in detail above. Nowhere in any of these sections is

there any teaching or suggestion of a file that describes the structure <u>and delete actions</u> for tables in a relational database. Moreover, since none of these sections of Ng teaches or suggests a file that describes the structure and delete actions for tables in a relational database, Ng cannot teach that the file is an Extensible Markup Language file, the file is generated based on user input to override default delete action identifiers in the file, or that the file is generated based on user input to insert one or more delete constraints in the file for one or more of the tables in the relational database. None of the references teaches or suggests these features.

Thus, dependent claims 21, 24-26, 28, 31-34, 36, 39-42, 47 and 48 also distinguish over Ng and Elmasri, either alone or in combination, by virtue of the specific features recited in these claims.

VII.   <u>35 U.S.C. § 103, Alleged Obviousness, Claims 22-23, 29-30 and 37-38</u>

The Office Action rejects claims 22-23, 29-30 and 37-38 under 35 U.S.C. § 103(a) as being unpatentable over Ng et al. (U.S. Patent No. 6,385,618 B1) in view of Text Book: Fundamentals of Database System (Third Edition) of Ramez Elmasri and Shamkant B. Navathe from Addison-Wisley and further in view of Crus et al. (U.S. Patent No. 4,947,320). This rejection is respectfully traversed for the same reasons stated in the previous Response filed March 24, 2004, the remarks of which are hereby incorporated by reference. This rejection is respectfully traversed.

The Office Action dated August 19, 2004 fails to provide rebuttal on Applicant's remarks with respect to claims 22-23, 29-30 and 37-38 from the Response dated March 24, 2004. In that Response, Applicant argued that the rejection is respectfully traversed for at least the same reasons as noted above with regard to claims 20, 27 and 35 from which claims 22-23, 29-30 and 37-38 depend. Specifically, Ng and Elmasri, taken alone or in combination, fail to teach or suggest determining a structure of a relational database by referring to a database meta-information class object and determining a delete action based on the structure described in the meta-information class object, or a class object that is generated based on a structure of a relational database and one or more delete actions for tables in the relational database.

Moreover, Crus does not provide for the deficiencies of Ng and Elmasri. Crus teaches a delete set null and a delete cascade operation, as discussed in previously filed Responses. However, Crus provides no teaching or suggestion regarding a class object that is generated based on a structure of a relational database and one or more delete actions for tables in the relational database. Crus also provides no teaching or suggestion regarding determining a structure of a relational database by involving a meta-information class object associated with the relational database and then determining a delete action based on the determined structure of the relational database. Thus, even if Crus were combinable with Ng and Elmasri, the result still would not be the invention recited in independent claims 20, 27 and 35, from which claims 22-23, 29-30 and 37-38 depend.

Furthermore, there is no teaching or suggestion in Crus to include information to identify a delete set null or delete cascade operation in a class object, as recited in claims 22, 29 and 37. While Crus may generally teach delete set null and delete cascade operations, there is nothing in Crus that teaches or suggests to include information regarding such operations in a class object generated based on a structure of a relational database and one or more delete actions.

In view of the above Applicant respectfully submits that the cited references, whether taken alone or in combination, fail to teach or suggest the features of independent claims 20, 27 and 35. At least by virtue of their dependency on claims 20, 27 and 35, the specific features of dependent claims 22-23, 29-30 and 37-38 are not taught or suggested by the cited references, either alone or in combination. Accordingly, Applicant respectfully requests withdrawal of the rejection of claims 22-23, 29-30 and 37-38.

VIII.   35 U.S.C. § 103, Alleged Obviousness, Claim 44

The Office Action rejects claim 44 under 35 U.S.C. § 103(a) as being unpatentable over Ng et al. (U.S. Patent No. 6,385,618 B1) in view of Text Book: Fundamentals of Database System (Third Edition) of Ramez Elmasri and Shamkant B. Navathe from Addison-Wisley and further in view of Sarkar (U.S. Patent 6,418,448 B1).

Because this rejection is essentially the same as in the Previous Office Action, this rejection is respectfully traversed for the same reasons stated in the previous Response filed March 24, 2004, the remarks of which are hereby incorporated by reference. This rejection is respectfully traversed.

The Office Action dated August 19, 2004 fails to provide rebuttal on Applicant's remarks with respect to claim 44 from the Response dated March 24, 2004. In that Response, Applicant argued that the features of claim 44 are similar to features previously discussed above. That is, claim 44 recites that the database meta-information generator class encapsulates information identifying a structure of a relational database and one or more delete actions into a class object. None of the cited art teaches or suggests a class object that encapsulates information identifying a structure of a relational database and one or more delete actions, as discussed at length above.
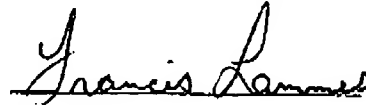
Therefore, dependent claim 44, which incorporates the features of independent claim 43, also defines over Ng, Elmasri and Sarkar at least by virtue of its dependency.

IX. Conclusion

It is respectfully urged that the subject application is patentable over the prior art of record and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

Respectfully submitted,

DATE: *November 16, 2004*

Francis Lammes

Francis Lammes
Reg. No. 55,353
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 385-8777
Agent for Applicant

Page 25 of 25
George – 09/544,274